Ogden Air Logistics Center

A Quality Process Performance Model for Software Development Projects

Using Monte Carlo Simulation to Predict Interim and Final Product Quality

David R. Webb

Senior Technical Program Manager

Hill Air Force Base, Utah

SSTC 2009

Process Quality

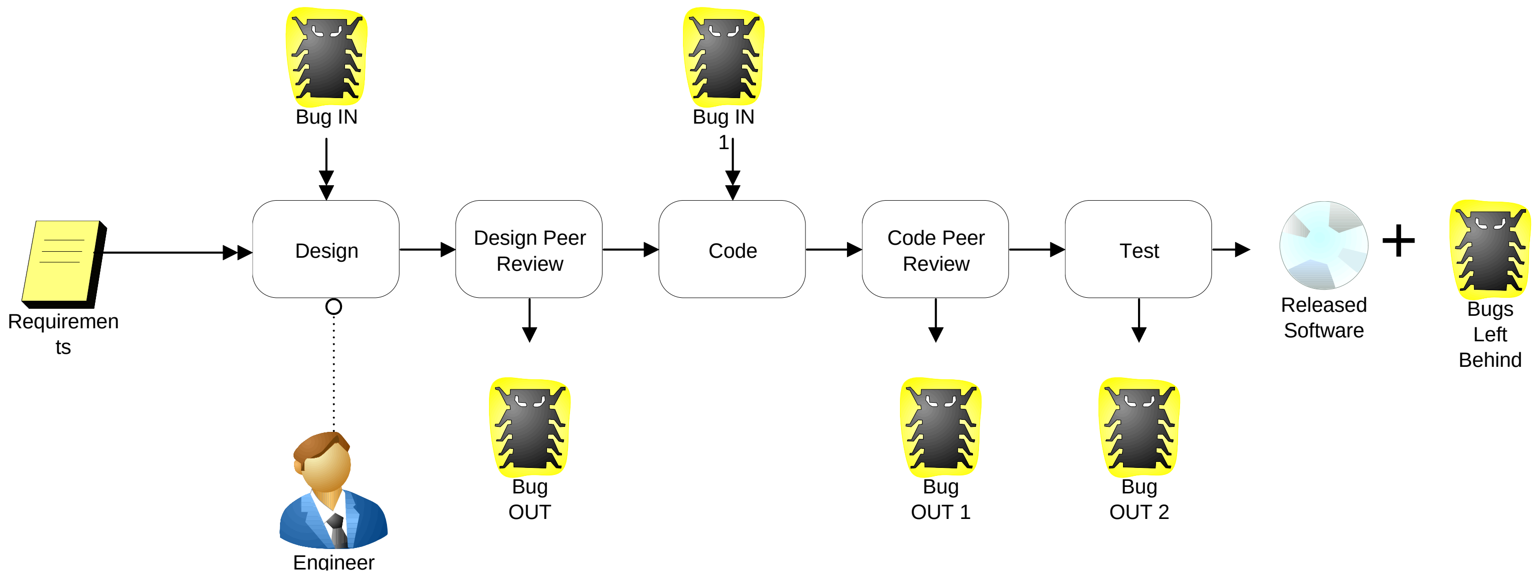
- **Focus on Defects**
  - **A defect is defined in the 520th Squadron Quality Management Plan as**

    "a product or product component that does not meet requirements or a design or implementation element that if not fixed could cause improper design, implementation, test, use or maintenance"
  - **The number of defects in the product is only one indication of product quality**
  - **Defects cause rework and become increasingly expensive to fix**
  - **Until we have functional software with relatively few defects, it doesn't make sense to focus too much on the other quality issues**

A Simple Quality Model

OGDEN AIR LOGISTICS CENTER

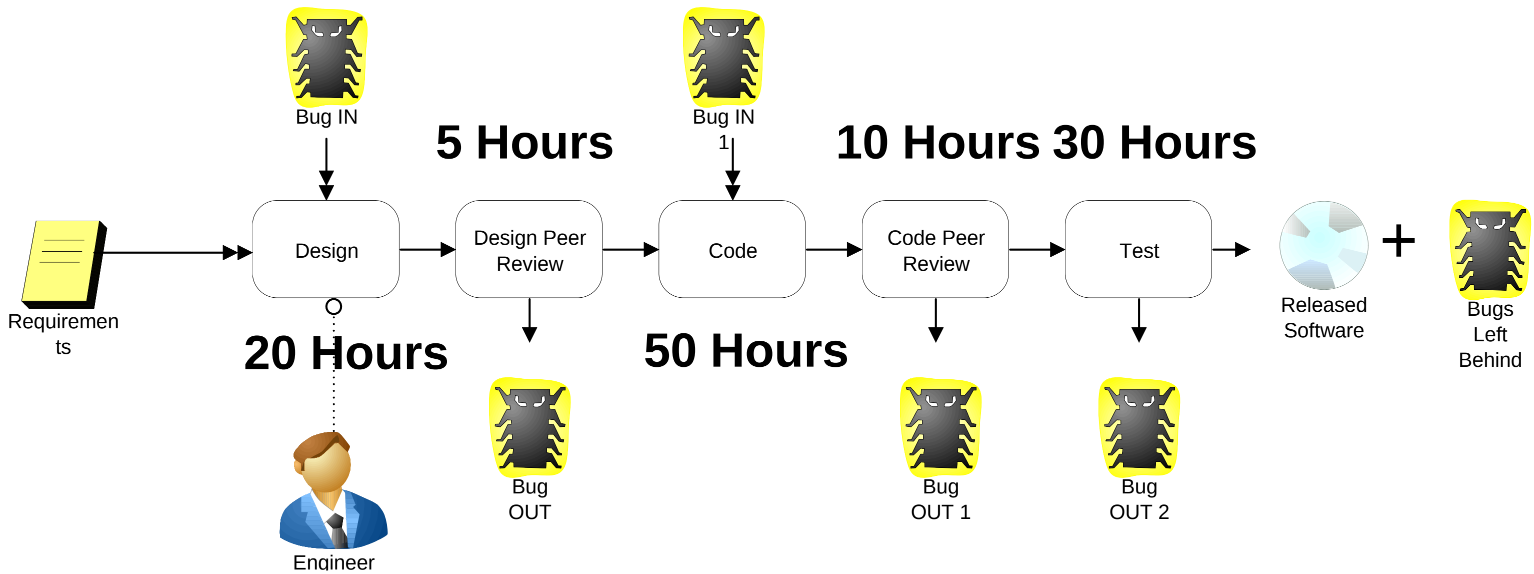
- Our processes have basically 2 kinds of defect-related activities:
  - Activities when defects are inadvertently injected
  - Activities when defects are sought for and removed



Estimating a Project

OGDEN AIR LOGISTICS CENTER

- **Effort and Schedule**
  - **Typically, we are able to estimate how long our schedule will take**
  - **We also typically break those estimates down into the phases our process – this becomes our WBS**



Gathering Historical Data – 1

OGDEN AIR LOGISTICS CENTER

- **Defect Injection Rate (DIR)**
  - **For all completed projects, we should examine all the defects found and determine during which phase of our process they were introduced**
  - **We also know, once the project is complete, how many hours were spent in those phases**
  - **DIR can be calculated as explained below:**

$$DIR_X = \frac{d_X}{aph_X}$$

d_X = defects injected in process block X
aph_X = actual cost performance in hours for block X

Gathering Historical Data – 2

- **Defect Detection Ratio (DDR)**
  - **As with DIR, we can examine closed projects to determine during which phases of our process defects were discovered**
  - **We also know, once the project is complete, how many total defects were found in each phase**
  - **DDR can be calculated as explained below:**

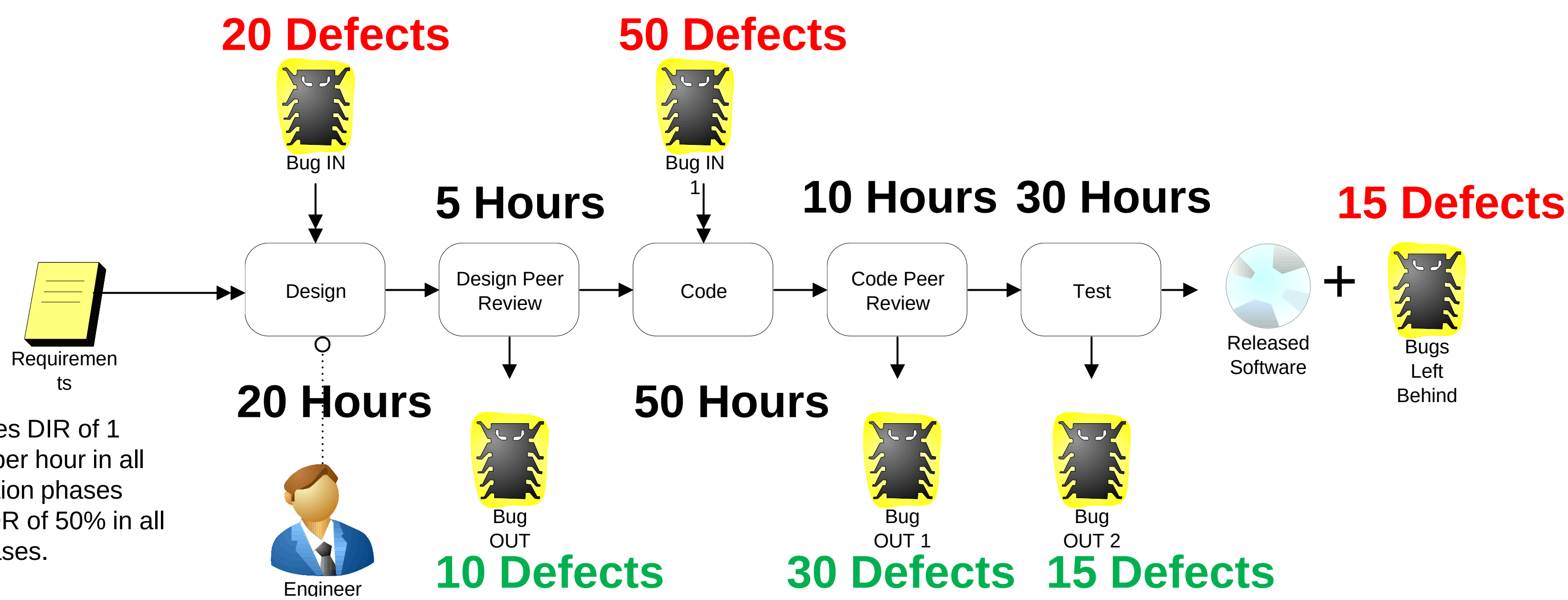
$$DDR_X = \frac{i_X}{i_X + e_X}$$

i_X = all defects found in QA activity for process block X

e_X = any defects injected in the process block(s) covered by QA activity X but detected at a later QA activity

Completing the Quality Model

- Defects Injected (DI)
  - Now that we know the DIR, we can use our hours estimate to project how many defects will be inadvertently injected in each production phase
- Defects Removed (DR)
  - Also, since we know the DDR of our QA phases, we can project how many of those defects will probably be removed
- Defects Remaining
  - Determining the bugs left behind is easy: Defects Remaining = DI - DR



Assumes DIR of 1 defect per hour in all production phases and DDR of 50% in all QA phases.

Quality Model Issues

- Effort Estimation
  - Productivity isn't always what you estimate it will be … sometimes you use more hours than planned, sometimes less.
- Quality Estimates
  - DIR can vary based upon team composition, the product being produced, the familiarity with the product and tools, etc.
  - DDR per phase varies based upon the same kinds of considerations.
- Updating the Model
  - The Model must take into account the variability of effort, defect injection and defect removal to be accurate

Accounting for Variability in Effort

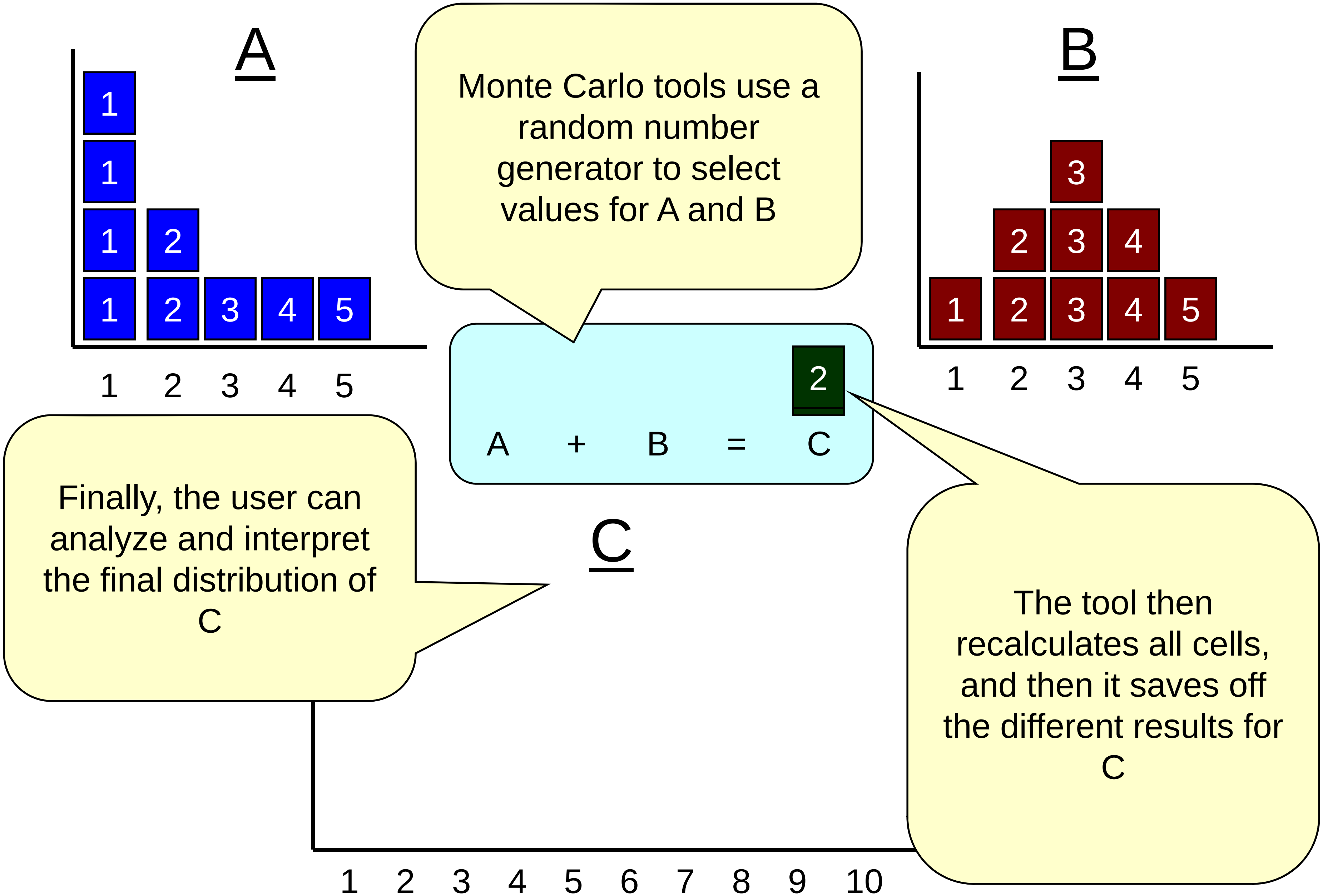
- **Effort Estimating**
  - **We can easily calculate a project's Cost Productivity Index (CPI) for historical projects**
  - **CPI is the ratio of planned to actual hours (or dollars)**
  - **We can divide our effort estimates by CPI to get a better estimate of what our real effort will be**
    - **A project that consistently overestimates will have a CPI > 1; dividing by the CPI will decrease the estimate**
    - **A project that consistently underestimates will have a CPI <1; dividing by the CPI will increase the estimate**
  - **However, just as CPI is not the same for every historical project, an average CPI may not be sufficient to properly adjust our effort estimates**

Accounting for Variability Using Monte Carol Simulation

- Monte Carlo Simulation
  - a technique using random numbers and probability distributions to solve problems
  - Uses "brute force" computational power to overcome situations where solving a problem analytically would be difficult
  - Iteratively applies the model hundreds or thousands of times to determine an expected solution
  - First extensively studied during the Manhattan project, where it was used to model neutron behavior

How Does Monte Carlo Work?

- **Monte Carlo Steps**
  1. **Create a parametric model**
  2. **Generate random inputs**
  3. **Evaluate the model and store the results**
  4. **Repeat steps 2 and 3 (x-1) more times**
  5. **Analyze the results of the x runs**


A
1
1
1 2
1 2 3 4 5
1 2 3 4 5
Monte Carlo tools use a random number generator to select values for A and B
B
3
2 3 4
1 2 3 4 5
1 2 3 4 5
2
A + B = C
Finally, the user can analyze and interpret the final distribution of C
C
The tool then recalculates all cells, and then it saves off the different results for C
1 2 3 4 5 6 7 8 9 10

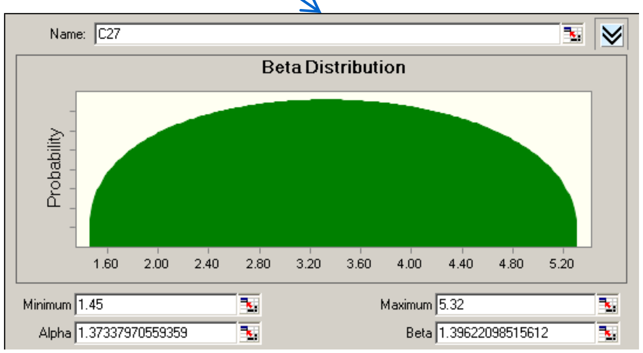
Applying Monte Carlo Simulation to the Quality Model

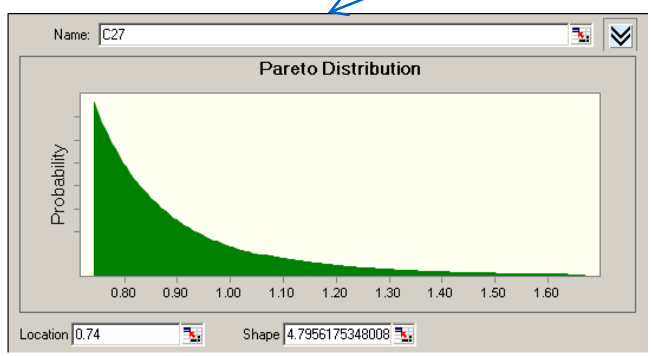
- Variability
  - Allow the following values to be variable
    - Cost Productivity Index
    - Defect Injection Rate per Phase
    - Defect Detection Ratio per Phase
  - Use Historical Data to Determine
    - Statistical distribution of data
    - Averages and limits of the data
- Apply Monte Carlo
  - Have the Monte Carlo tool run the model thousands of times
  - Each time, Monte Carlo will choose a random value for CPI, DIRs and DDRs, generating a new result
  - Over time, a profile will be built showing the distribution of likely outcomes

Historical Variability

	DIR Design	DDR Design PR	DIR Code	DDR Code PR	DDR Unit Test	DDR System Test	DDR Acceptance Test	CPI
Project 1	3	60%	10	80%	50%	45%	10%	0.80
Project 2	4	58%	12	50%	45%	55%	12%	1.20
Project 3	2.5	62%	9	75%	65%	45%	5%	0.78
Project 4	3.6	75%	11	50%	52%	65%	7%	0.80
Project 5	4.2	80%	8	60%	66%	45%	8%	1.32
Project 6	1.8	43%	12	65%	52%	55%	5%	1.02
Project 7	2	55%	15	75%	53%	45%	6%	1.00
Project 8	5	88%	6	70%	47%	68%	8%	0.80
Project 9	4	47%	8	60%	52%	72%	9%	0.92
Project 10	2.8	78%	7.5	55%	56%	47%	12%	0.80
Project 11	3.6	52%	10	65%	59%	62%	6%	0.79
Project 12	4	60%	12	75%	68%	42%	8%	1.25
Project 13	5	65%	16	80%	66%	45%	23%	0.75
Project 14	2	75%	11	45%	54%	39%	7%	0.80
Project 15	3	70%	9	55%	50%	45%	5%	0.88
Averages	3.37	65%	10.43	64%	56%	52%	9%	0.93

Note: Only two distributions are show … there are similar distributions for each column





Setting Up the Monte Carlo Simulation

OGDEN AIR LOGISTICS CENTER

Quality Model for Development Projects

	DIR	DDR	Est. Hours	CPI	Defects Injected	Defects Removed
Planning	0	0%	50	1	0	0
Design	3.37	0%	120	1	404.4	0
Design Peer Review	0	65%	20	1	0	262.86
Code	10.43	0%	200	1	2086	0
Code Peer Review	0	64%	40	1	0	1425.6256
Unit Test	0	56%	80	1	0	449.072064
System Test	0	52%	40	1	0	183.4780147
Acceptance Test	0	9%	40	1	0	15.2427889
Release	0	0%	15	1	0	0

Totals	2490.4	2336.28

Remaining Defects	154.12

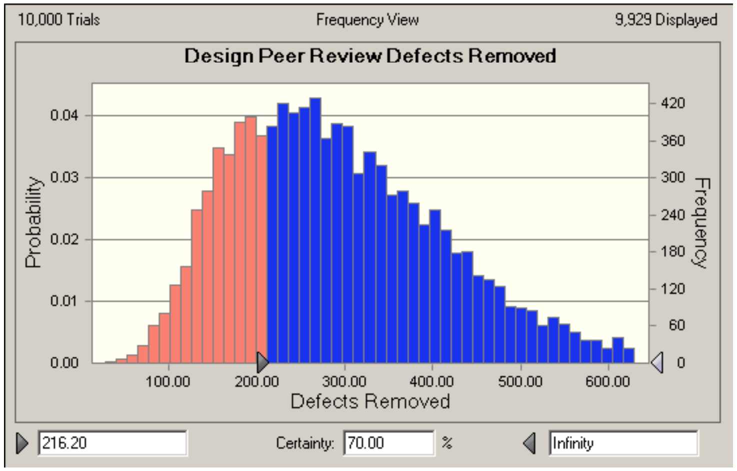
Interim Results for each Monte Carlo run

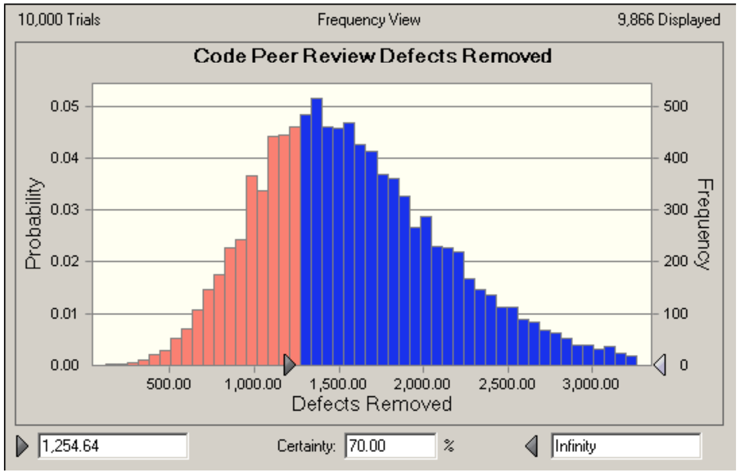
Final Results for each Monte Carlo run

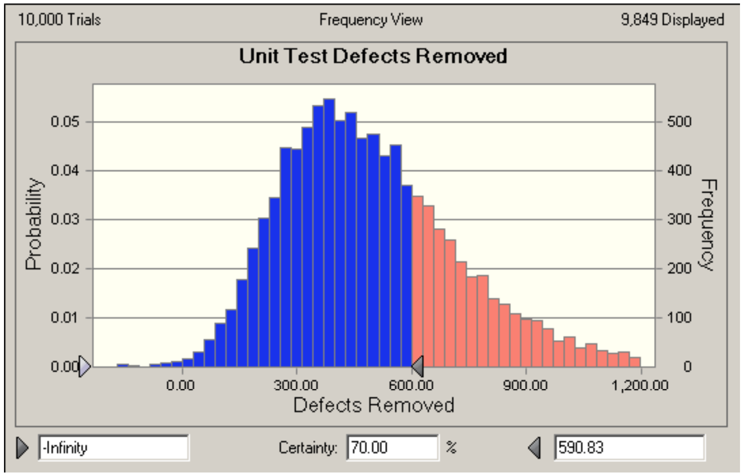
Variable Model Inputs Based upon the distributions of the historical data

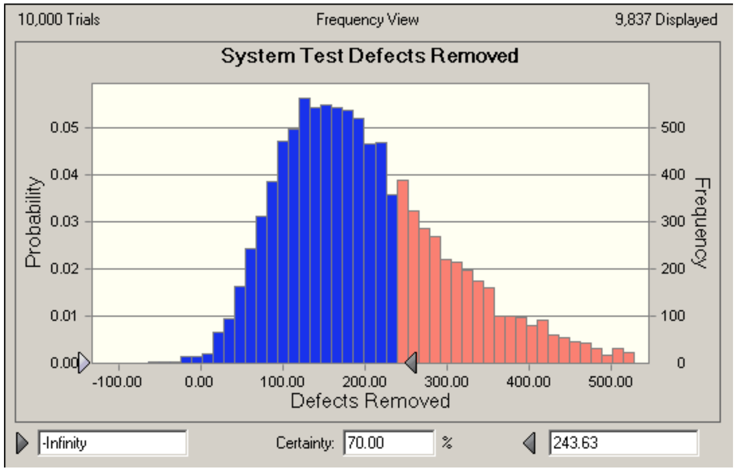
Some Interim Results (70% Certainty)

OGDEN AIR LOGISTICS CENTER



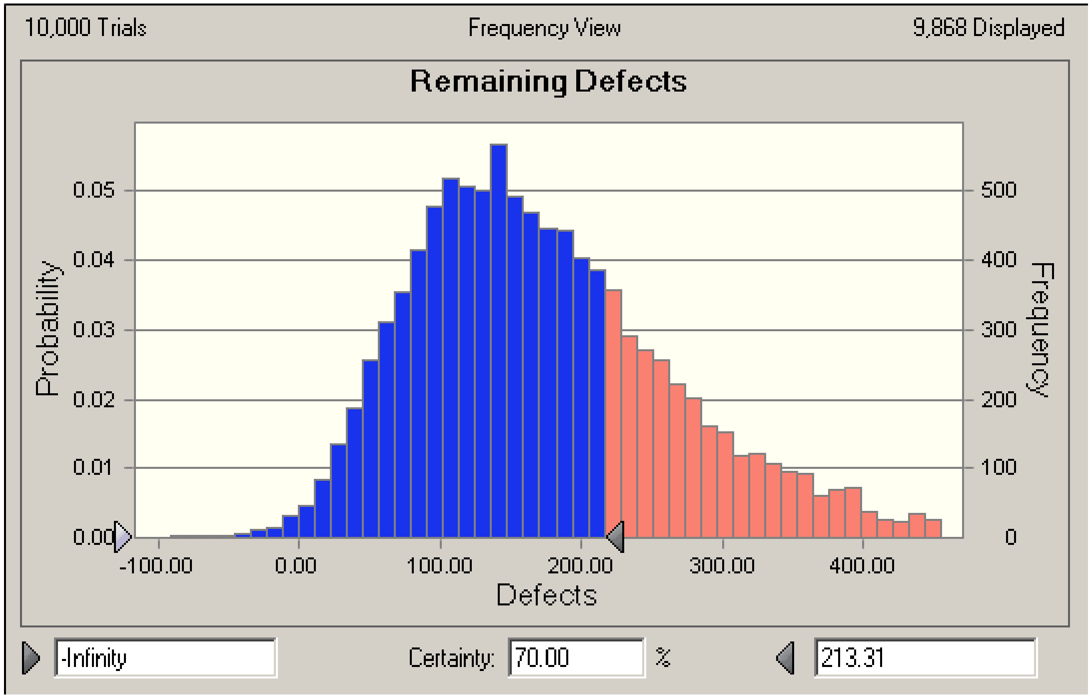






Final Results of the Quality Model (70% Certainty)

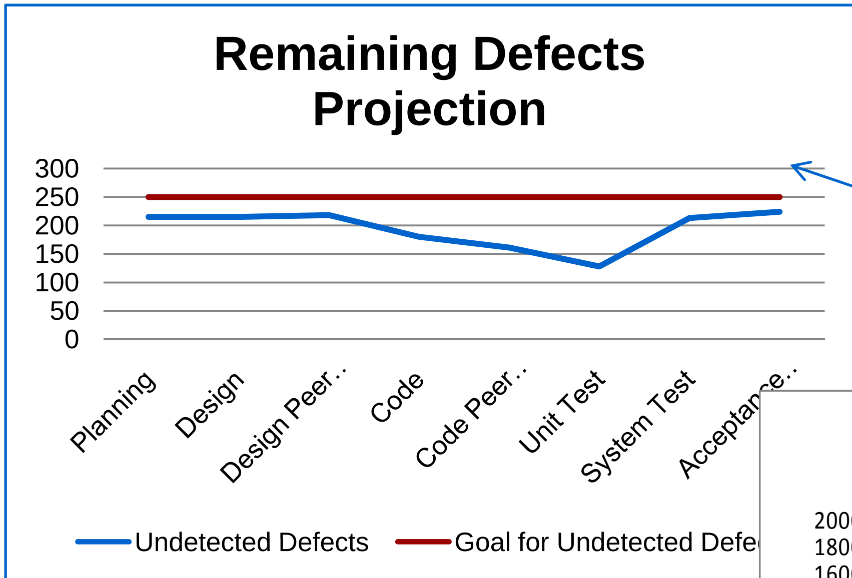
OGDEN AIR LOGISTICS CENTER



Tracking the Project Using the Model

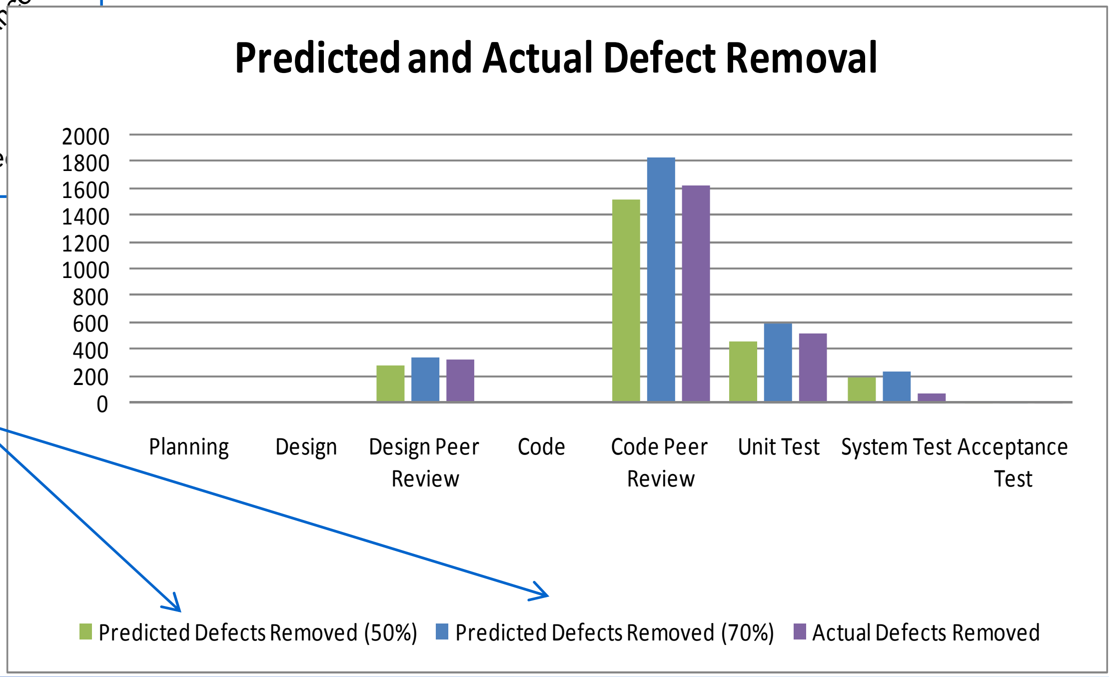
- **During Planning**
  - **Run the model**
  - **Determine projects for final outcome and all interim outcomes**
  - **Compare the final outcome to project goals**
  - **If goals are not met, then a process improvement is warranted (i.e., changes to increase DDR or decrease DIR)**
- **During Project Execution**
  - **Compare the interim results to actual results**
  - **The model will only tell you the MAXIMUM number you should expect within your certainty level**
  - **If you see results lower than that number, you're probably OK**
  - **If you see results much higher that that number, then you need to do some investigation**
  - **Once you have true interim results, replace the Monte Carlo variation with the real numbers and re-run the model – do you still have a final outcome that meets project goals?**

Examples of Tracking the Project



Determined by re-running the model as actual data replace the Monte Carlo Simulation estimates.

Note that you can vary the certainty levels you want to look at. You may wish to look at higher certainty levels for planning, lower levels to set “stretch goals”



Model Demonstration

Summary

OGDEN AIR LOGISTICS CENTER

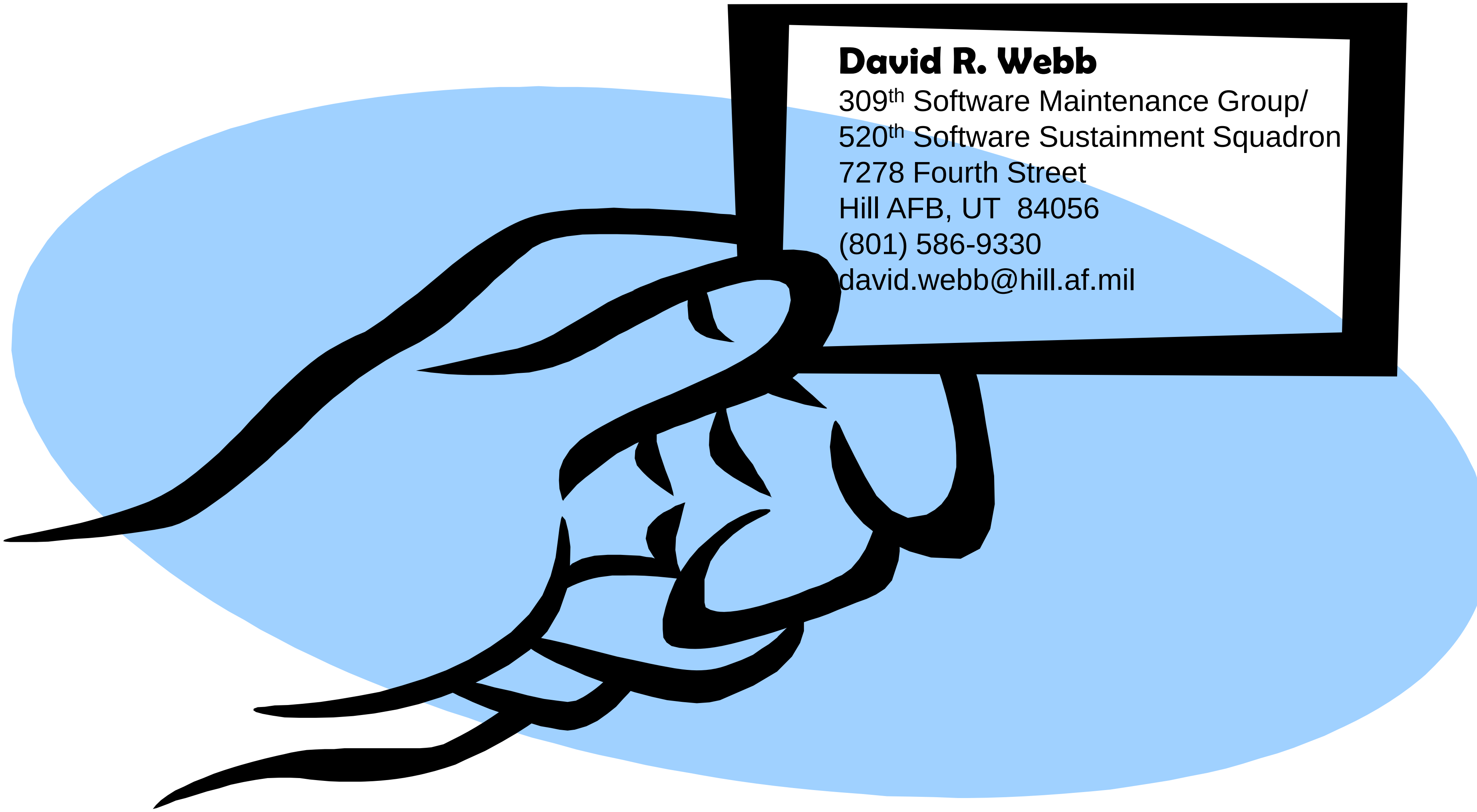
- The Software Maintenance Group at Hill Air Force Base has created a Quality Model applicable for most software development projects
- Quality is modeled by predicting defect injection and removal using historical data
- Variation is taken into account by using a Monte Carlo Simulation to adjust estimates, defect injection rates and defect detection ratios
- Interim results can be used to guide the project toward a final quality goal
- Actual data replaces projected in the model as the project progresses

Questions

TICS CENTER

Contact Information

OGDEN AIR LOGISTICS CENTER



David R. Webb
309th Software Maintenance Group/
520th Software Sustainment Squadron
7278 Fourth Street
Hill AFB, UT 84056
(801) 586-9330
david.webb@hill.af.mil